Refine Search

Search Results -

Terms	Documents
L11 and (JUMP or JMP)	25

Database:

US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search:

```
L12
```

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Monday, March 14, 2005 Printable Copy Create Case

Set Name side by side	**Query**	**Hit Count**	**Set Name** result set
	DB=USPT; PLUR=NO; OP=OR		
L12	L11 and (JUMP or JMP)	25	L12
L11	L10 and l9	204	L11
L10	L8 AND l2	204	L10
L9	L8 AND l3	269	L9
L8	717/$$$.ccls OR 713/$$$.ccls.	13568	L8
L7	L6 AND l5	0	L7
L6	L4 AND monitor.ab.	15	L6
L5	L4 AND security.ab.	15	L5
L4	L3 and ((byte ADJ code) OR (bytecode))	525	L4
L3	java and monitor	4280	L3
L2	java and security and monitor	1831	L2
L1	java near security near monitor	0	L1

END OF SEARCH HISTORY

Hit List

Clear Generate Collection Print Fwd Refs Bkwd Refs
Generate OACS

Search Results - Record(s) 1 through 25 of 25 returned.

□ 1. Document ID: US 6668325 B1

```
L12: Entry 1 of 25                    File: USPT                    Dec 23, 2003

US-PAT-NO: 6668325
DOCUMENT-IDENTIFIER: US 6668325 B1
** See image for Certificate of Correction **

TITLE: Obfuscation techniques for enhancing software security

DATE-ISSUED: December 23, 2003

INVENTOR-INFORMATION:
NAME                          CITY        STATE   ZIP CODE    COUNTRY
Collberg; Christian Sven      Auckland                        NZ
Thomborson; Clark David       Auckland                        NZ
Low; Douglas Wai Kok          Auckland                        NZ

US-CL-CURRENT: 713/194; 713/200

ABSTRACT:

The present invention provides obfuscation techniques for enhancing software
security. In one embodiment, a method for obfuscation techniques for enhancing
software security includes selecting a subset of code (e.g., compiled source code
of an application) to obfuscate, and obfuscating the selected subset of the code.
The obfuscating includes applying an obfuscating transformation to the selected
subset of the code. The transformed code can be weakly equivalent to the
untransformed code. The applied transformation can be selected based on a desired
level of security (e.g., resistance to reverse engineering). The applied
transformation can include a control transformation that can be creating using
opaque constructs, which can be constructed using aliasing and concurrency
techniques. Accordingly, the code can be obfuscated for enhanced software security
based on a desired level of obfuscation (e.g., based on a desired potency,
resilience, and cost).

171 Claims, 55 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 27
```

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

□ 2. Document ID: US 6643775 B1

L12: Entry 2 of 25 File: USPT Nov 4, 2003

US-PAT-NO: 6643775
DOCUMENT-IDENTIFIER: US 6643775 B1

TITLE: Use of code obfuscation to inhibit generation of non-use-restricted versions of copy protected software applications

DATE-ISSUED: November 4, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Granger; Mark J.	Azusa	CA		
Smith; Cyrus E.	Monrovia	CA		
Hoffman; Matthew I.	South Pasadena	CA		

US-CL-CURRENT: 713/190; 380/255, 380/268, 713/201

ABSTRACT:

Three methods are disclosed for protecting software applications from unauthorized distribution and use (piracy). The first method involves using values generated by a conventional ESD (Electronic Security Device) to encrypt and/or decrypt user data (such as a file) that is generated and used by the application. In a preferred embodiment, the user data is encrypted (such as during a write to memory) using values returned by the ESD, and the user data is later decrypted using like values returned by a software-implemented ESD simulator. The second and third methods involve the use of special development tools that make the task of analyzing the application's copy protection code (such as the code used to encrypt and/or decrypt user data) significantly more difficult. Specifically, the second method involves using pseudocode to implement some or all of the application's copy protection functions. The pseudocode for a given function is generated (preferably in encrypted form) from actual code using a special development tool, and is then imbedded within the application together with a corresponding pseudocode interpreter. The interpreter fetches, decrypts and executes the pseudocode when the function is called. Because no disassemblers or other development tools exist for analyzing the pseudocode, the task of analyzing the copy protection functions becomes significantly more complex. The third method involves the use of a special obfuscation tool to convert the code for selected copy-protection functions into unnecessarily long, inefficient sequences of machine code. In one implementation of the obfuscation tool, the developer can control the quantity of code that is generated by specifying one or more control parameters. The three methods can also be used to protect software license management systems from security attacks.

48 Claims, 14 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 11

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

□ 3. Document ID: US 6480959 B1

L12: Entry 3 of 25 File: USPT Nov 12, 2002

US-PAT-NO: 6480959
DOCUMENT-IDENTIFIER: US 6480959 B1

TITLE: Software system and associated methods for controlling the use of computer programs

DATE-ISSUED: November 12, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Granger; Mark J.	Azusa	CA		
Smith; Cyrus E.	Monrovia	CA		
Hoffman; Matthew I.	South Pasadena	CA		

US-CL-CURRENT: 713/189; 713/200

ABSTRACT:

Three methods are disclosed for protecting software applications from unauthorized distribution and use (piracy). The first method involves using values generated by a conventional ESD (Electronic Security Device) to encrypt and/or decrypt user data (such as a file) that is generated and used by the application. In a preferred embodiment, the user data is encrypted (such as during a write to memory) using values returned by the ESD, and the user data is later decrypted using like values returned by a software-implemented ESD simulator. The second and third methods involve the use of special development tools that make the task of analyzing the application's copy protection code (such as the code used to encrypt and/or decrypt user data) significantly more difficult. Specifically, the second method involves using pseudocode to implement some or all of the application's copy protection functions. The pseudocode for a given function is generated (preferably in encrypted form) from actual code using a special development tool, and is then imbedded within the application together with a corresponding pseudocode interpreter. The interpreter fetches, decrypts and executes the pseudocode when the function is called. Because no disassemblers or other development tools exist for analyzing the pseudocode, the task of analyzing the copy protection functions becomes significantly more complex. The third method involves the use of a special obfuscation tool to convert the code for selected copy-protection functions into unnecessarily long, inefficient sequences of machine code. In one implementation of the obfuscation tool, the developer can control the quantity of code that is generated by specifying one or more control parameters. The three methods can also be used to protect software license management systems from security attacks.

50 Claims, 14 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 11

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

□ 4. Document ID: US 6460141 B1

L12: Entry 4 of 25 File: USPT Oct 1, 2002

US-PAT-NO: 6460141
DOCUMENT-IDENTIFIER: US 6460141 B1

TITLE: Security and access management system for web-enabled and non-web-enabled applications and content on a computer network

DATE-ISSUED: October 1, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Olden; Eric M.	San Francisco	CA		

US-CL-CURRENT: 713/201; 713/202

ABSTRACT:

A security and access management system provides unified access management to address the specific problems facing the deployment of security for the Web and non-Web environment. Unified access management consists of strategic approaches to unify all key aspects of Web and non-Web security policies, including access control, authorization, authentication, auditing, data privacy, administration, and business rules. Unified access management also addresses technical scalability requirements needed to successfully deploy a reliable unified Web and non-Web security system. The security and access management system provides the technology required to support these key factors as they relate to Web and non-Web security. The security and access management system operates in combination with network and system security tools such as firewalls, network intrusion detection tools, and systems management tools to provide comprehensive security for the Web-enabled enterprise.

3 Claims, 37 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 36

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

□ 5. Document ID: US 6427140 B1

L12: Entry 5 of 25 File: USPT Jul 30, 2002

US-PAT-NO: 6427140
DOCUMENT-IDENTIFIER: US 6427140 B1
**** See image for Certificate of Correction ****

TITLE: Systems and methods for secure transaction management and electronic rights protection

DATE-ISSUED: July 30, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Ginter; Karl L.	Beltsville	MD		
Shear; Victor H.	Bethseda	MD		
Spahn; Francis J.	El Cerrito	CA		
Van Wie; David M.	Sunnyvale	CA		

US-CL-CURRENT: 705/80; 705/53, 713/193

ABSTRACT:

The present invention provides systems and methods for secure transaction management and electronic rights protection. Electronic appliances such as computers equipped in accordance with the present invention help to ensure that information is accessed and used only in authorized ways, and maintain the integrity, availability, and/or confidentiality of the information. Such electronic appliances provide a distributed virtual distribution environment (VDE) that may enforce a secure chain of handling and control, for example, to control and/or meter or otherwise monitor use of electronically stored or disseminated information. Such a virtual distribution environment may be used to protect rights of various participants in electronic commerce and other electronic or electronic-facilitated transactions. Distributed and other operating systems, environments and architectures, such as, for example, those using tamper-resistant hardware-based processors, may establish security at each node. These techniques may be used to support an all-electronic information distribution, for example, utilizing the "electronic highway."

30 Claims, 155 Drawing figures
Exemplary Claim Number: 24
Number of Drawing Sheets: 146

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

□ 6. Document ID: US 6415316 B1

L12: Entry 6 of 25 File: USPT Jul 2, 2002

US-PAT-NO: 6415316
DOCUMENT-IDENTIFIER: US 6415316 B1
**** See image for Certificate of Correction ****

TITLE: Method and apparatus for implementing a web page diary

DATE-ISSUED: July 2, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Van Der Meer; Joannes Jozef Everardus	Amersfoort			NL

US-CL-CURRENT: 709/203; 705/8, 709/217, 713/166, 713/167, 715/501.1, 715/513, 715/526

ABSTRACT:

A method and apparatus to create a "diary" containing multimedia references to contents of Websites. These references (also called addresses) can be to, for example, text, bookmarks, images, programs, movies, etc. Many content objects are provided via the Websites of "content providers," with the specific intent of making the content objects available to a user to place in his diary. Each diary page has a format specified by a cover. The cover is provided by a cover provider and specifies where on the diary page the diary owner can place his content. The

name "diary" arises because the invention preferably allows the user to save these references in association with dates and/or times. The pages of a user's diary may be navigated like a book, moving forward and backward through the pages or jumping to a particular page. In addition to storing references to Web information, the user can also jot down reminders, enter appointments, and birthdays, etc. for dates. A user is allowed to choose a visual "theme" for the pages of his diary. This theme can be changed at any time by the user and reflects how the user wants to present himself and his diary to the world. The user can set various levels of privacy for different portions of his diary.

39 Claims, 33 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 23

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

□ 7. Document ID: US 6378075 B1

L12: Entry 7 of 25 File: USPT Apr 23, 2002

US-PAT-NO: 6378075
DOCUMENT-IDENTIFIER: US 6378075 B1

TITLE: Trusted agent for electronic commerce

DATE-ISSUED: April 23, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Goldstein; Theodore C.	Palo Alto	CA		
Martinez; Ronald G.	San Francisco	CA		
Rubin; Paul	Milpitas	CA		

US-CL-CURRENT: 713/200; 705/64

ABSTRACT:

A trusted agent server provides a networked application that assists a customer in managing their online commercial affairs. A user contacts the server using a network access device, such as a browser on a personal computer. The trusted agent client component augments the user's network access device to perform business transactions on behalf of the user. The user controls these transactions through the trusted agent server. A trusted agent service is a trusted agent client component application which operates in conjunction with the trusted agent server. The trusted agent service is an Internet-based mechanism that makes single-click buying available on any commercial Web site. The trusted agent also provides customers with access to personal and credit card information used during single-click transactions, smart receipts used for ongoing customer support, merchant and product preference settings, and direct response product offerings keyed to these preferences. Because this information is all stored on the trusted agent server, it is available to any device connected to the Internet. The trusted agent service is implemented by operating the trusted agent server.

22 Claims, 9 Drawing figures

Exemplary Claim Number: 1
Number of Drawing Sheets: 9

□ 8. Document ID: US 6374286 B1

L12: Entry 8 of 25 File: USPT Apr 16, 2002

US-PAT-NO: 6374286
DOCUMENT-IDENTIFIER: US 6374286 B1

TITLE: Real time processor capable of concurrently running multiple independent JAVA machines

DATE-ISSUED: April 16, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Gee; John K.	Mt. Vernon	IA		
Greve; David A.	Cedar Rapids	IA		
Hardin; David S.	Cedar Rapids	IA		
Mass; Allen P.	Lisbon	IA		
Masters; Michael H.	Cedar Rapids	IA		
Mykris; Nick M.	Cedar Rapids	IA		
Wilding; Matthew M.	Cedar Rapids	IA		

US-CL-CURRENT: 718/108; 710/260, 713/502, 718/1

ABSTRACT:

Multiple Java Virtual Machines (JVMs) operate on a single direct execution JAVA processor with each JVM operating in a separate time slice called a partition. Each JVM has its own data and control structures and is assigned a fixed area of memory. Each partition is also allotted a fixed period of time in which to operate, and, at the end of the allotted time, a context switch is forced to another JVM operating in the next partition. The context switch does not transfer control directly from one JVM to another JVM. Instead, at the end of a partition time period control is switched from the currently operating JVM to a "master JVM" during a time period called an "interstice." The master JVM handles system interrupts and housekeeping duties. At the end of the interstice time period, the master JVM starts a proxy thread associated with the next JVM to become operational. The proxy thread handles JVM-specific interrupts and checks the status of the associated JVM. If the JVM appears operational the proxy thread transfers control to the JVM thread. Time intervals such as partition times and interstice times are enforced by hardware timers and memory accesses are checked by address comparison circuitry to prevent a system failure due to a malfunction in either the master JVM or another JVM.

25 Claims, 23 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 21

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

□ 9. Document ID: US 6363488 B1

L12: Entry 9 of 25 File: USPT Mar 26, 2002

US-PAT-NO: 6363488
DOCUMENT-IDENTIFIER: US 6363488 B1
**** See image for Certificate of Correction ****

TITLE: Systems and methods for secure transaction management and electronic rights protection

DATE-ISSUED: March 26, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Ginter; Karl L.	Beltsville	MD		
Shear; Victor H.	Bethesda	MD		
Spahn; Francis J.	El Cerrito	CA		
Van Wie; David M.	Eugene	OR		

US-CL-CURRENT: 713/201; 705/14, 705/53

ABSTRACT:

The present invention provides systems and methods for secure transaction management and electronic rights protection. Electronic appliances such as computers equipped in accordance with the present invention help to ensure that information is accessed and used only in authorized ways, and maintain the integrity, availability, and/or confidentiality of the information. Such electronic appliances provide a distributed virtual distribution environment (VDE) that may enforce a secure chain of handling and control, for example, to control and/or meter or otherwise monitor use of electronically stored or disseminated information. Such a virtual distribution environment may be used to protect rights of various participants in electronic commerce and other electronic or electronic-facilitated transactions. Distributed and other operating systems, environments and architectures, such as, for example, those using tamper-resistant hardware-based processors, may establish security at each node. These techniques may be used to support an all-electronic information distribution, for example, utilizing the "electronic highway."

6 Claims, 155 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 146

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

□ 10. Document ID: US 6353892 B1

L12: Entry 10 of 25 File: USPT Mar 5, 2002

US-PAT-NO: 6353892
DOCUMENT-IDENTIFIER: US 6353892 B1
**** See image for Certificate of Correction ****

TITLE: Copy protection of digital images transmitted over networks

DATE-ISSUED: March 5, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Schreiber; Daniel	Beit Shemesh			IL
Goldman; Andrew	Beit Shemesh			IL

US-CL-CURRENT: 713/201

ABSTRACT:

A method for protecting digital images distributed over a network, including the steps of receiving a request from a client computer running a network browser, for an original layout page containing references to digital images therein, parsing the original layout page for the references to digital images, generating a modified layout page from the original layout page by replacing at least one of the references to digital images in the original layout page with references to substitute data, and sending the modified layout page to the client computer. A system is also described and claimed.

32 Claims, 19 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 15

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

□ 11. Document ID: US 6334189 B1

L12: Entry 11 of 25 File: USPT Dec 25, 2001

US-PAT-NO: 6334189
DOCUMENT-IDENTIFIER: US 6334189 B1

TITLE: Use of pseudocode to protect software from unauthorized use

DATE-ISSUED: December 25, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Granger; Mark J.	Azusa	CA		
Smith; Cyrus E.	Monrovia	CA		
Hoffman; Matthew I.	South Pasadena	CA		

US-CL-CURRENT: 713/200; 380/255, 380/268, 713/201

ABSTRACT:

Three methods are disclosed for protecting software applications from unauthorized distribution and use (piracy). The first method involves using values generated by a conventional ESD (Electronic Security Device) to encrypt and/or decrypt user data (such as a file) that is generated and used by the application. In a preferred embodiment, the user data is encrypted (such as during a write to memory) using values returned by the ESD, and the user data is later decrypted using like values returned by a software-implemented ESD simulator. The second and third methods involve the use of special development tools that make the task of analyzing the application's copy protection code (such as the code used to encrypt and/or decrypt user data) significantly more difficult. Specifically, the second method involves using pseudocode to implement some or all of the application's copy protection functions. The pseudocode for a given function is generated (preferably in encrypted form) from actual code using a special development tool, and is then imbedded within the application together with a corresponding pseudocode interpreter. The interpreter fetches, decrypts and executes the pseudocode when the function is called. Because no disassemblers or other development tools exist for analyzing the pseudocode, the task of analyzing the copy protection functions becomes significantly more complex. The third method involves the use of a special obfuscation tool to convert the code for selected copy-protection functions into unnecessarily long, inefficient sequences of machine code. In one implementation of the obfuscation tool, the developer can control the quantity of code that is generated by specifying one or more control parameters. The three methods can also be used to protect software license management systems from security attacks.

30 Claims, 14 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 11

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

□ 12. Document ID: US 6327661 B1

L12: Entry 12 of 25 File: USPT Dec 4, 2001

US-PAT-NO: 6327661
DOCUMENT-IDENTIFIER: US 6327661 B1

TITLE: Using unpredictable information to minimize leakage from smartcards and other cryptosystems

DATE-ISSUED: December 4, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Kocher; Paul C.	San Francisco	CA		
Jaffe; Joshua M.	San Francisco	CA		
Jun; Benjamin C.	Palo Alto	CA		

US-CL-CURRENT: 713/193; 380/28, 380/46, 380/47, 713/322, 713/323, 713/501

ABSTRACT:

Methods and apparatuses are disclosed for securing cryptosystems against external monitoring attacks by reducing the amount (and signal to noise ratio) of useful information leaked during processing. This is generally accomplished by incorporating unpredictable information into the cryptographic processing. Various embodiments of the invention use techniques such as reduction of signal to noise ratios, random noise generation, clock skipping, and introducing entropy into the order of processing operations or the execution path. The techniques may be implemented in hardware or software, may use a combination of digital and analog techniques, and may be deployed in a variety of cryptographic devices.

36 Claims, 2 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 2

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

□ 13. Document ID: US 6298446 B1

L12: Entry 13 of 25 File: USPT Oct 2, 2001

US-PAT-NO: 6298446
DOCUMENT-IDENTIFIER: US 6298446 B1

TITLE: Method and system for copyright protection of digital images transmitted over networks

DATE-ISSUED: October 2, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Schreiber; Daniel	Beit Shemesh			IL
Goldman; Andrew	Beit Shemesh			IL

US-CL-CURRENT: 713/201; 713/200

ABSTRACT:

A method for protecting digital images distributed over a network, including the steps of receiving a request from a client computer running a network browser, for an original layout page containing references to digital images therein, parsing the original layout page for the references to digital images, generating a modified layout page from the original layout page by replacing at least one of the references to digital images in the original layout page with references to substitute data, and sending the modified layout page to the client computer. A system is also described and claimed.

12 Claims, 19 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 14

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

□ 14. Document ID: US 6282573 B1

L12: Entry 14 of 25 File: USPT Aug 28, 2001

US-PAT-NO: 6282573
DOCUMENT-IDENTIFIER: US 6282573 B1

TITLE: Computer architecture for managing courseware in a shared use operating environment

DATE-ISSUED: August 28, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Darago; Vincent S.	Manasquan	NJ		
Jenkins; Christopher	Springville	UT		

US-CL-CURRENT: 709/229; 709/217, 713/201

ABSTRACT:

Methods, devices, and systems are provided in a multi-level computer architecture which provides improved capabilities for managing courseware and other content in a shared use operating environment such as a computer network. In particular, the invention provides a commercial networked instruction content delivery method and system which does not exclude synchronous sharing but is focused on asynchronous sharing. Security in the architecture provides content property holders with the ability to know how many minutes of use an individual made of licensed material and with increased certainty that their material cannot be used, copied, or sold in usable form unless lo and until a user site is connected or reconnected to a minute-by-minute counter which is located off the premises of the user. This security link helps protect software and other works which are being sold or licensed to an individual, organization, or entity, and creates income opportunities for owners of such content.

37 Claims, 7 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

□ 15. Document ID: US 6275938 B1

L12: Entry 15 of 25 File: USPT Aug 14, 2001

US-PAT-NO: 6275938
DOCUMENT-IDENTIFIER: US 6275938 B1

TITLE: Security enhancement for untrusted executable code

DATE-ISSUED: August 14, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bond; Barry	Renton	WA		
Bharati; Sudeep	Bellevue	WA		

US-CL-CURRENT: 713/200; 713/201, 713/202, 714/38, 714/53, 714/55

ABSTRACT:

Untrusted executable code programs (applets or controls) are written in native, directly executable code. The executable code is loaded into a pre-allocated memory range (sandbox) from which references to outside memory are severely restricted by checks (sniff code) added to the executable code. Conventional application-program interface (API) calls in the untrusted code are replaced with translation-code modules (thunks) that allow the executable code to access the host operating system, while preventing breaches of the host system's security. Static links in the code are replaced by calls to thunk modules. When an API call is made during execution, control transfers to the thunk, which determines whether the API call is one which should be allowed to execute on the operating system.

20 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

□ 16. Document ID: US 6253326 B1

L12: Entry 16 of 25 File: USPT Jun 26, 2001

US-PAT-NO: 6253326
DOCUMENT-IDENTIFIER: US 6253326 B1

TITLE: Method and system for secure communications

DATE-ISSUED: June 26, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Lincke; Scott D.	San Carlos	CA		
Marianetti, II; Ronald	Morgan Hill	CA		

US-CL-CURRENT: 713/201; 380/255, 380/270, 713/168, 713/200

ABSTRACT:

A communications system and methods for securely transmitting a message between a wireless client and a proxy server are provided. A method for transmitting a message from the wireless client to a proxy server is provided. The message includes at least one packet of data and is encrypted using a data encryption key. The data encryption key is encrypted using a proxy server public key prior to sending the encrypted data encryption key to the proxy server. A method for transmitting a message from the proxy server to the wireless client is also provided. The proxy server recovers the data encryption key using the proxy server private key corresponding to the proxy server public key. The proxy server encrypts the message using the data encryption key and transmits the encrypted message to the wireless client. A communications system for secure communications comprising a source of data, a proxy server and a wireless client is also provided. Each transaction in the communications system comprises at least one request message and at least one response message. For each transaction, the wireless client encrypts a data encryption key using a proxy server public key. Messages exchanged between the wireless client and the proxy server are encrypted using the transaction specific data encryption key.

36 Claims, 14 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 14

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

□ 17. Document ID: US 6199181 B1

L12: Entry 17 of 25 File: USPT Mar 6, 2001

US-PAT-NO: 6199181
DOCUMENT-IDENTIFIER: US 6199181 B1
**** See image for Certificate of Correction ****

TITLE: Method and system for maintaining restricted operating environments for application programs or operating systems

DATE-ISSUED: March 6, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Rechef; Eran	Lehavim			IL
Raanan; Gil	Zoran			IL
Solan; Eilon	Herzlia			IL

US-CL-CURRENT: 714/38; 713/201

ABSTRACT:

A method for protecting an operating environment on a processor from a rogue program operating on the processor comprising isolating simultaneously executing programs or operating systems is disclosed. Memory space for use only by the first program while the first program is executing is allocated. Communication between the first program and the computer's operating environment is accomplished through a single link employing one of several methods including using shared memory space, a dedicated interrupt or a dedicated I/O port. The monitor manages a restricted operating environment for the first program on the processor, the restricted operating environment preventing the first program from accessing resources on the processor except for the allocated memory space the single communication link.

21 Claims, 10 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 9

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

□ 18. Document ID: US 6023764 A

L12: Entry 18 of 25 File: USPT Feb 8, 2000

US-PAT-NO: 6023764
DOCUMENT-IDENTIFIER: US 6023764 A

TITLE: Method and apparatus for providing security certificate management for Java Applets

DATE-ISSUED: February 8, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Curtis; Bryce Allen	Round Rock	TX		

US-CL-CURRENT: 713/200

ABSTRACT:

The present invention defines a means for establishing a secure connection between a Java Applet and a secure web server for protocols other than Https via the use of a Java Security Service. More specifically, the present invention uses the web browser's installed certificates to setup and establish an encrypted session between the Java Applet and the secure web server. The secure connection is then used to retrieve the certificates required by the Java security service.

15 Claims, 5 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

□ 19. Document ID: US 6009540 A

L12: Entry 19 of 25 File: USPT Dec 28, 1999

US-PAT-NO: 6009540
DOCUMENT-IDENTIFIER: US 6009540 A

TITLE: Logic module for implementing system changes on PC architecture computers

DATE-ISSUED: December 28, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Craft; Thomas W.	Laguna Hills	CA		
Dobbs; Donald Lee	Mission Viejo	CA		

US-CL-CURRENT: 714/30; 713/400, 713/500, 713/503, 714/54, 714/55, 714/731

ABSTRACT:

A system, method and apparatus including a logic module, preferably embodied as an electronic card that operates in combination with a PC to correct errors caused by deficiencies existing in logic residing on the PC's motherboard, such as the PC's BIOS. The preferred logic card includes a transceiver module, a memory module (e.g. an EPROM or Masked ROM) containing storage elements and executable code stored as pages. The preferred logic card also includes a page register module in communication with the transceiver and the memory, and a paging mechanism that cooperates with the page register and the transceiver for allowing only a predetermined number of bytes (pages) of executable code to be accessible for operation in the PC's main-memory in order to correct errors caused by deficiencies existing in logic residing on the PC's motherboard.

9 Claims, 10 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 10

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

□ 20. Document ID: US 5982891 A

L12: Entry 20 of 25 File: USPT Nov 9, 1999

US-PAT-NO: 5982891
DOCUMENT-IDENTIFIER: US 5982891 A

TITLE: Systems and methods for secure transaction management and electronic rights protection

DATE-ISSUED: November 9, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Ginter; Karl L.	Beltsville	MD		
Shear; Victor H.	Bethesda	MD		
Spahn; Francis J.	El Cerrito	CA		
Van Wie; David M.	Sunnyvale	CA		

US-CL-CURRENT: 705/54; 705/26, 713/167

ABSTRACT:

The present invention provides systems and methods for secure transaction management and electronic rights protection. Electronic appliances such as computers equipped in accordance with the present invention help to ensure that information is accessed and used only in authorized ways, and maintain the integrity, availability, and/or confidentiality of the information. Such electronic appliances provide a distributed virtual distribution environment (VDE) that may enforce a secure chain of handling and control, for example, to control and/or meter or otherwise monitor use of electronically stored or disseminated

information. Such a virtual distribution environment may be used to protect rights of various participants in electronic commerce and other electronic or electronic-facilitated transactions. Distributed and other operating systems, environments and architectures, such as, for example, those using tamper-resistant hardware-based processors, may establish security at each node. These techniques may be used to support an all-electronic information distribution, for example, utilizing the "electronic highway."

102 Claims, 153 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 146

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

□ 21. Document ID: US 5974549 A

L12: Entry 21 of 25 File: USPT Oct 26, 1999

US-PAT-NO: 5974549
DOCUMENT-IDENTIFIER: US 5974549 A

TITLE: Security monitor

DATE-ISSUED: October 26, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Golan; Gilad	Ramat Hasharon			IL

US-CL-CURRENT: 713/200; 714/47

ABSTRACT:

The present invention is a method of creating a secure sandbox within which a plurality of downloaded software components can execute in a secure manner. The software components can be of any type, e.g., Java, ActiveX, Netscape plugin, etc. The invention implements a security monitor that is injected to the address space of an arbitrary monitored application such as a Web browser, e.g., Internet Explorer, Netscape Navigator, etc. The monitored application then executes in a secure mode in which every software component downloaded executes in a secure sandbox. The security monitor detects when such a software component is downloaded and is operative to create the sandbox around it before it is permitted to execute. If the software component attempts to commit an action that breaches security, it halts the software component's execution and issues a warning to the user. The security monitor detects attempted security breaches by the software component in accordance with a user configurable security policy. Such a policy may include limiting file read/write access, access to directories, disk access, creation and the reading/writing of network connections, access to system resources and services and access to the address spaces of other processes.

19 Claims, 15 Drawing figures
Exemplary Claim Number: 12
Number of Drawing Sheets: 14

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

□ 22. Document ID: US 5917912 A

L12: Entry 22 of 25 File: USPT Jun 29, 1999

US-PAT-NO: 5917912
DOCUMENT-IDENTIFIER: US 5917912 A

TITLE: System and methods for secure transaction management and electronic rights protection

DATE-ISSUED: June 29, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Ginter; Karl L.	Beltsville	MD		
Shear; Victor H.	Bethesda	MD		
Spahn; Francis J.	El Cerrito	CA		
Van Wie; David M.	Sunnyvale	CA		

US-CL-CURRENT: 713/187; 705/40, 713/164, 719/312

ABSTRACT:

The present invention provides systems and methods for secure transaction management and electronic rights protection. Electronic appliances such as computers equipped in accordance with the present invention help to ensure that information is accessed and used only in authorized ways, and maintain the integrity, availability, and/or confidentiality of the information. Such electronic appliances provide a distributed virtual distribution environment (VDE) that may enforce a secure chain of handling and control, for example, to control and/or meter or otherwise monitor use of electronically stored or disseminated information. Such a virtual distribution environment may be used to protect rights of various participants in electronic commerce and other electronic or electronic-facilitated transactions. Distributed and other operating systems, environments and architectures, such as, for example, those using tamper-resistant hardware-based processors, may establish security at each node. These techniques may be used to support an all-electronic information distribution, for example, utilizing the "electronic highway."

58 Claims, 153 Drawing figures
Exemplary Claim Number: 58
Number of Drawing Sheets: 146

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

□ 23. Document ID: US 5915019 A

L12: Entry 23 of 25 File: USPT Jun 22, 1999

US-PAT-NO: 5915019
DOCUMENT-IDENTIFIER: US 5915019 A

TITLE: Systems and methods for secure transaction management and electronic rights protection

DATE-ISSUED: June 22, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Ginter; Karl L.	Beltsville	MD		
Shear; Victor H.	Bethesda	MD		
Spahn; Francis J.	El Cerrito	CA		
Van Wie; David M.	Sunnyvale	CA		

US-CL-CURRENT: 705/54; 705/26, 705/400, 713/200

ABSTRACT:

The present invention provides systems and methods for secure transaction management and electronic rights protection. Electronic appliances such as computers equipped in accordance with the present invention help to ensure that information is accessed and used only in authorized ways, and maintain the integrity, availability, and/or confidentiality of the information. Such electronic appliances provide a distributed virtual distribution environment (VDE) that may enforce a secure chain of handling and control, for example, to control and/or meter or otherwise monitor use of electronically stored or disseminated information. Such a virtual distribution environment may be used to protect rights of various participants in electronic commerce and other electronic or electronic-facilitated transactions. Distributed and other operating systems, environments and architectures, such as, for example, those using tamper-resistant hardware-based processors, may establish security at each node. These techniques may be used to support an all-electronic information distribution, for example, utilizing the "electronic highway."

101 Claims, 155 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 146

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

□ 24. Document ID: US 5892900 A

L12: Entry 24 of 25 File: USPT Apr 6, 1999

US-PAT-NO: 5892900
DOCUMENT-IDENTIFIER: US 5892900 A
**** See image for Certificate of Correction ****

TITLE: Systems and methods for secure transaction management and electronic rights protection

DATE-ISSUED: April 6, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Ginter; Karl L.	Beltsville	MD		
Shear; Victor H.	Bethesda	MD		
Sibert; W. Olin	Lexington	MA		
Spahn; Francis J.	El Cerrito	CA		
Van Wie; David M.	Sunnyvale	CA		

US-CL-CURRENT: 713/200; 713/201

ABSTRACT:

The present invention provides systems and methods for electronic commerce including secure transaction management and electronic rights protection. Electronic appliances such as computers employed in accordance with the present invention help to ensure that information is accessed and used only in authorized ways, and maintain the integrity, availability, and/or confidentiality of the information. Secure subsystems used with such electronic appliances provide a distributed virtual distribution environment (VDE) that may enforce a secure chain of handling and control, for example, to control and/or meter or otherwise monitor use of electronically stored or disseminated information. Such a virtual distribution environment may be used to protect rights of various participants in electronic commerce and other electronic or electronic-facilitated transactions. Secure distributed and other operating system environments and architectures, employing, for example, secure semiconductor processing arrangements that may establish secure, protected environments at each node. These techniques may be used to support an end-to-end electronic information distribution capability that may be used, for example, utilizing the "electronic highway."

220 Claims, 177 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 163

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

□ 25. Document ID: US 5870544 A

L12: Entry 25 of 25 File: USPT Feb 9, 1999

US-PAT-NO: 5870544
DOCUMENT-IDENTIFIER: US 5870544 A

TITLE: Method and apparatus for creating a secure connection between a java applet and a web server

DATE-ISSUED: February 9, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Curtis; Bryce Allen	Round Rock	TX		

US-CL-CURRENT: 713/201; 709/229, 713/150, 713/151, 713/156, 713/200

ABSTRACT:

The present invention defines a a method, an apparatus and a computer program product for establishing a secure connection between a Java Applet and a secure web server for protocols other than Https via the use of a Java Security Service. More specifically, the present invention uses the web browser's installed certificates to setup and establish an encrypted session between the Java Applet and the secure web server. The secure connection is then used to retrieve the certificates required by the Java security service.

6 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Terms	Documents
L11 and (JUMP or JMP)	25

Display Format: REV Change Format

Previous Page Next Page Go to Doc#

Hit List

Clear Generate Collection Print Fwd Refs Bkwd Refs
Generate OACS

Search Results - Record(s) 1 through 15 of 15 returned.

☐ 1. Document ID: US 6850989 B1

```
  L6: Entry 1 of 15                     File: USPT                    Feb 1, 2005
```

US-PAT-NO: 6850989
DOCUMENT-IDENTIFIER: US 6850989 B1

TITLE: Method and apparatus for automatically configuring a network switch

DATE-ISSUED: February 1, 2005

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Lavian; Tal I.	Sunnyvale	CA		
Lau; Stephen	Milpitas	CA		
Ong; Lyndon Y.	San Jose	CA		

US-CL-CURRENT: 709/242; 709/224, 709/249

ABSTRACT:

A method and apparatus for automatically configuring a network switch having external network data ports, a processor, and memory. Network data is monitored on the external network data port. Information about the network data traffic is compared to one or more threshold conditions. The network switch is automatically configured if the network data meets one of the threshold conditions. The monitor and configuration functions can be performed by software running on the processor which has been downloaded from an external network maintenance station through a maintenance data port. Information about the network data traffic can be uploaded to the external network maintenance station through a maintenance data port.

17 Claims, 15 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 9

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

☐ 2. Document ID: US 6802054 B2

```
  L6: Entry 2 of 15                     File: USPT                    Oct 5, 2004
```

US-PAT-NO: 6802054
DOCUMENT-IDENTIFIER: US 6802054 B2

TITLE: Generation of runtime execution traces of applications and associated problem determination

DATE-ISSUED: October 5, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Faraj; Mazen	North York			CA

US-CL-CURRENT: 717/128; 714/45

ABSTRACT:

A computer system for generating and analyzing application trace data includes a monitor for launching Java language virtual machines using the Java Platform Debug Architecture to enable the virtual machines to generate event data on the occurrence of specified events during application execution on the virtual machines. The event data is placed on an event queue and the monitor removes the event data from the event queue for forwarding to a logging service. The logging service records the event data in a trace file. A set of problem determination tools use defined product description, and the trace file data to provide an analysis to a user based on a defined level of analysis selected by the user from product, component, code and logical levels of analysis.

25 Claims, 1 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 1

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

□ 3. Document ID: US 6792601 B1

L6: Entry 3 of 15 File: USPT Sep 14, 2004

US-PAT-NO: 6792601
DOCUMENT-IDENTIFIER: US 6792601 B1

TITLE: Multiple mode object locking method and system

DATE-ISSUED: September 14, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Dimpsey; Robert Tod	Travis	TX		
Hoflich; Benjamin Joseph	Austin	TX		
Peacock; Brian David	North Baddlesley			GB

US-CL-CURRENT: 718/102; 710/200, 711/205, 711/206, 718/101, 718/103, 718/104

ABSTRACT:

An object-based multi-threaded computing system has a cyclic garbage collection strategy and includes an object locking system having (i) a first mode in which access by a single thread without contention to an object is controlled by a monitor internal to said object, and (ii) a second mode in which access by multiple threads with contention to said object is controlled by a monitor external to said object. For any given object a transition from the first mode to the second mode is termed inflation, and a transition from the second mode to the first mode is termed deflation. Responsive to the start of a period of contention for an object in said first mode, the object is inflated to the second mode, and an inflation rate counter is incremented. After the period of contention has concluded the value of the inflation rate counter is compared against a predetermined value in order to determine whether or not to deflate the object. The inflation rate counter is reset at every garbage collection cycle.

20 Claims, 5 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

□ 4. Document ID: US 6769022 B1

L6: Entry 4 of 15 File: USPT Jul 27, 2004

US-PAT-NO: 6769022
DOCUMENT-IDENTIFIER: US 6769022 B1

TITLE: Methods and apparatus for managing heterogeneous storage devices

DATE-ISSUED: July 27, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
DeKoning; Rodney A.	Augusta	KS		
Delaney; William P.	Wichita	KS		
Jantz; Ray M.	Wichita	KS		
Weber; Bret S.	Wichita	KS		
Courtright, II; William V.	Pittsburgh	PA		

US-CL-CURRENT: 709/223; 709/201, 709/203, 709/224, 709/246, 710/18, 714/26, 714/31

ABSTRACT:

A system for monitoring and managing devices on network comprising one or more managed devices connected to the network and storage means for storing a device management application program associated with each of the managed devices. The system further includes a management station which is in communication with each of the managed devices across the network, and the management station is in communication with the storage means. When a user wishes to monitor, configure, or manage one of the managed devices on the network, the user preferably selects the managed device to be managed and the management station retrieves from the storage means the device management application program associated with the selected managed device. By the management station processing the management application program for the selected managed device, the management station allows the user to

monitor the status of the managed device, as well as change the configuration of and fix errors with the managed device.

22 Claims, 16 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 13

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

□ 5. Document ID: US 6735760 B1

L6: Entry 5 of 15 File: USPT May 11, 2004

US-PAT-NO: 6735760
DOCUMENT-IDENTIFIER: US 6735760 B1

TITLE: Relaxed lock protocol

DATE-ISSUED: May 11, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Dice; David	Foxborough	MA		

US-CL-CURRENT: 717/139; 717/140, 717/151, 718/102

ABSTRACT:

An object-oriented compiler/interpreter allocates monitor records for use in implementing synchronized operations on objects. When a synchronization operation is to be performed on an object, a thread that is to perform the operation "inflates" the object's monitor by placing into its header a pointer to the monitor record as well as an indication of the monitor's inflated status. When a thread is to release its lock on an object, it first consults a reference-count field in the monitor record to determine whether any other threads are synchronized on the object. It then dissociates the object from the monitor record. The dissociation is not atomic with the reference-count check, so the releasing thread checks the reference count again. If that count indicates that further objects had employed the monitor record to synchronize on the object in the interim, then the unlocking thread wakes all waiting threads.

45 Claims, 22 Drawing figures
Exemplary Claim Number: 12
Number of Drawing Sheets: 20

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

□ 6. Document ID: US 6654949 B1

L6: Entry 6 of 15 File: USPT Nov 25, 2003

US-PAT-NO: 6654949
DOCUMENT-IDENTIFIER: US 6654949 B1
**** See image for Certificate of Correction ****

TITLE: Method and system for monitoring the execution of hybrid source code

DATE-ISSUED: November 25, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Fraenkel; Michael L.	Raleigh	NC		
Gerken; Christopher H.	Apex	NC		
Ryman; Arthur G.	Thornhill			CA
Yu; Patsy S. H.	Toronto			CA
Yuen; Siu C.	Scarborough			CA

US-CL-CURRENT: 717/130; 717/124, 717/127

ABSTRACT:

This invention describes a system and method for monitoring the execution of hybrid source code such as JavaServer Pages (JSP) code. The system comprises a page compiler, which is called by a server for translating JSP code into a servlet for execution by the server. The page compiler during translation of the JSP code inserts instrumentation in the compiled JSP code for supporting execution tracing by an execution monitor. The execution monitor receives outputs from the page compiler, the servlet and the raw JSP code for displaying selected information about the execution of the JSP code to the developer on a graphical user interface. The execution monitor thus allows the developer to view the correlation between the JSP code, the servlet code and the HTML code that is generated by the servlet.

24 Claims, 6 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 6

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

□ 7. Document ID: US 6557168 B1

L6: Entry 7 of 15 File: USPT Apr 29, 2003

US-PAT-NO: 6557168
DOCUMENT-IDENTIFIER: US 6557168 B1

TITLE: System and method for minimizing inter-application interference among static synchronized methods

DATE-ISSUED: April 29, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Czajkowski; Grzegorz J.	Mountain View	CA		

US-CL-CURRENT: 717/151; 713/375, 717/127

ABSTRACT:

A system and method for isolating the execution of a plurality of applications. A plurality of monitors are provided for a plurality of applications to access a static synchronized method. The applications are enabled to call the static synchronized method concurrently by accessing the static synchronized method through the plurality of monitors. A plurality of threads within one of the applications are excluded from calling the static synchronized method concurrently. The source code or bytecode for the synchronized method may be transformed by removing a method-level monitor and adding the plurality of monitors inside the method. In one embodiment, each static synchronized method is replaced with a corresponding static non-synchronized method. The applications may be further isolated by placing the static fields of shared classes into a static field class, which has one instance per utilizing application. The static non-synchronized method includes the body of the corresponding static synchronized method, wherein the body is synchronized on the instance of the static field class that corresponds to the utilizing application.

30 Claims, 11 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 11

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

☐ 8. Document ID: US 6532531 B1

L6: Entry 8 of 15 File: USPT Mar 11, 2003

US-PAT-NO: 6532531
DOCUMENT-IDENTIFIER: US 6532531 B1

TITLE: Method frame storage using multiple memory circuits

DATE-ISSUED: March 11, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
O'Connor; James Michael	Mountain View	CA		
Tremblay; Marc	Palo Alto	CA		

US-CL-CURRENT: 712/202; 711/6, 712/217, 718/1

ABSTRACT:

A memory architecture in accordance with an embodiment of the present invention improves the speed of method invocation. Specifically, method frames of method calls are stored in two different memory circuits. The first memory circuit stores the execution environment of each method call, and the second memory circuit stores parameters, variables or operands of the method calls. In one embodiment the execution environment includes a return program counter, a return frame, a return constant pool, a current method vector, and a current monitor address. In some embodiments, the memory circuits are stacks; therefore, the stack management unit

to cache can be used to cache either or both memory circuits. The stack management unit can include a stack cache to accelerate data transfers between a stack-based computing system and the stacks. In one embodiment, the stack management unit includes a stack cache, a dribble manager unit, and a stack control. The dribble manager unit include fill control it and a spill control unit. Since the vast majority of memory accesses to the stack occur at or near the top of the stack, the dribble manager unit maintains the top portion of the stack in the stack cache. When the stack-based computing system is popping data off of the stack and a fill condition occurs, the fill control unit transfer data from the stack to the bottom of the stack cache to maintain the top portion of the stack in the stack cache. Typically, a fill condition occurs as the stack cache becomes empty and a spill condition occurs as the stack cache becomes full.

56 Claims, 17 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 17

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

□ 9. Document ID: US 6510352 B1

L6: Entry 9 of 15 File: USPT Jan 21, 2003

US-PAT-NO: 6510352
DOCUMENT-IDENTIFIER: US 6510352 B1

TITLE: Methods and apparatus for object-based process control

DATE-ISSUED: January 21, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Badavas; Paul C.	Southboro	MA		
Hansen; Peter D.	Wellesley	MA		

US-CL-CURRENT: 700/19; 700/18, 700/20, 700/48, 700/49, 700/50, 700/52, 718/100

ABSTRACT:

The provides improved control devices, systems and methods for operation thereof. These rely on control devices that provide virtual machine environments in which Java objects, or other such software constructs, are executed to implement control (e.g., to monitor and/or control a device, process or system). These objects define blocks which are the basic functional unit of the control. They also define the input, output and body parts from which blocks are formed, and the signals that are communicated between blocks. The objects also define nested and composite groupings of blocks used to control loops and higher-level control functions.

72 Claims, 18 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 18

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

□ 10. Document ID: US 6480901 B1

L6: Entry 10 of 15 File: USPT Nov 12, 2002

US-PAT-NO: 6480901
DOCUMENT-IDENTIFIER: US 6480901 B1

TITLE: System for monitoring and managing devices on a network from a management station via a proxy server that provides protocol converter

DATE-ISSUED: November 12, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Weber; Bret S.	Wichita	KS		
DeKoning; Rodney A.	Augusta	KS		
Delaney; William P.	Wichita	KS		
Jantz; Ray M.	Wichita	KS		
Courtright, II; William V.	Pittsburgh	PA		

US-CL-CURRENT: 709/246; 709/223

ABSTRACT:

A system and method for monitoring and managing devices on a network. The system and method preferably comprises a proxy server connected to the network and a managed device connected to the proxy server. The system further comprises storage means for storing a device management application program associated with the managed device, and a management station in communication with the managed device via the proxy server and in communication with the storage means. The management station preferably is configured to retrieve the device management application program from the storage means and process the device management application program. As the management station processes the device management application program, the management station is able to monitor and manage the managed device. In particular, the management station can send management commands to a controller of the managed device via the proxy server, and the management station can receive notifications from the managed device, also via the proxy server.

2 Claims, 16 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 13

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

□ 11. Document ID: US 6327700 B1

L6: Entry 11 of 15 File: USPT Dec 4, 2001

US-PAT-NO: 6327700

DOCUMENT-IDENTIFIER: US 6327700 B1

TITLE: Method and system for identifying instrumentation targets in computer programs related to logical transactions

DATE-ISSUED: December 4, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Chen; J. Bradley	Seattle	WA		
Bershad; Brian N.	Seattle	WA		

US-CL-CURRENT: 717/127; 717/130

ABSTRACT:

A method and system for identifying sets of instructions within a computer program, execution of which serve as an indicator for processing of a transaction by the computer program and that together comprise a witness set. The witness set may be employed to monitor execution of the computer program and detect processing of the transaction. Witness sets are constructed by iteratively filtering an initial set of instructions based on profile data collected during execution of the computer program.

28 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

□ 12. Document ID: US 6314563 B1

L6: Entry 12 of 15 File: USPT Nov 6, 2001

US-PAT-NO: 6314563
DOCUMENT-IDENTIFIER: US 6314563 B1
**** See image for Certificate of Correction ****

TITLE: Expedited object locking and unlocking

DATE-ISSUED: November 6, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Agesen; Ole	Needham	MA		
Detlefs; David L.	Westford	MA		
Garthwaite; Alex	Philadelphia	PA		

US-CL-CURRENT: 717/108; 717/116, 717/139, 717/151

ABSTRACT:

An object structure's header (40) allocates a two-bit synchronization-state field (42) solely to monitor data for implementing synchronization on that object. When the object is locked by a particular execution thread, or when one or more execution threads are waiting for a lock or notification on that object, its header contains a pointer to monitor resources in the form of a linked list of lock records (50, 52, 54) associated with the threads involved. The synchronization-state field (42) ordinarily contains an indication of whether such a linked list exists and, if so, whether its first member is associated with a thread that has a lock on the object. When a thread attempts to gain access to that linked list, it employs an atomic swap operation to place a special busy value in that lock-state field (42) and write its execution-environment pointer into the object's header (40). If the previous value of that field was not the special busy value, the thread uses the header's previous contents to perform its intended synchronization operation. Otherwise, it obtains that information through its own execution environment (44, 46, or 48) or that of the thread whose identifier the object header previously contained. When the thread completes its synchronization operation, it employs an atomic compare-and-swap operation to write the results into the object's header if that header still contains the thread identifier that the thread originally wrote there. Otherwise, it communicates that information to its successor thread if the thread identifier is different and thereby indicates that at least one successor is contending for access to the linked list.

68 Claims, 35 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 28

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

□ 13. Document ID: US 6175868 B1

L6: Entry 13 of 15 File: USPT Jan 16, 2001

US-PAT-NO: 6175868
DOCUMENT-IDENTIFIER: US 6175868 B1

TITLE: Method and apparatus for automatically configuring a network switch

DATE-ISSUED: January 16, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Lavian; Tal I.	Sunnyvale	CA		
Lau; Stephen	Milpitas	CA		
Ong; Lyndon Y.	San Jose	CA		

US-CL-CURRENT: 709/223; 709/232, 709/238

ABSTRACT:

A method and apparatus for automatically configuring a network switch having external network data ports, a processor, and memory. Network data is monitored on the external network data port. Information about the network data traffic is compared to one or more threshold conditions. The network switch is automatically configured if the network data meets one of the threshold conditions. The monitor and configuration functions can be performed by software running on the processor which has been downloaded from an external network maintenance station through a maintenance data port. Information about the network data traffic can be uploaded to the external network maintenance station through a maintenance data port.

15 Claims, 15 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 9

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

□ 14. Document ID: US 6173442 B1

L6: Entry 14 of 15 File: USPT Jan 9, 2001

US-PAT-NO: 6173442
DOCUMENT-IDENTIFIER: US 6173442 B1

TITLE: Busy-wait-free synchronization

DATE-ISSUED: January 9, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Agesen; Ole	Needham	MA		
Detlefs; David L.	Westford	MA		
Garthwaite; Alex	Philadelphia	PA		
Knippel; Ross C.	Half Moon Bay	CA		
Ramakrishna; Y. Srinivas	Mountain View	CA		
White; Derek	Reading	MA		

US-CL-CURRENT: 717/141; 707/8, 717/127, 717/162

ABSTRACT:

An object structure's header (40) allocates a two-bit synchronization-state field (42) solely to monitor data for implementing synchronization on that object. When the object is locked by a particular execution thread, or when one or more execution threads are waiting for a lock or notification on that object, its header contains a pointer to monitor resources in the form of a linked list of lock records (50, 52, 54) associated with the threads involved. The synchronization-state field (42) ordinarily contains an indication of whether such a linked list exists and, if so, whether its first member is associated with a thread that has a lock on the object. When a thread attempts to gain access to that linked list, it employs an atomic swap operation to place a special busy value in that lock-state field (42) and write its execution-environment pointer into the object's header (40). If the previous value of that field was not the special busy value, the thread uses the header's previous contents to perform its intended synchronization operation. Otherwise, it obtains that information through its own execution environment (44, 46, or 48) or that of the thread whose identifier the object header previously contained. When the thread completes its synchronization operation, it employs an atomic compare-and-swap operation to write the results into the object's header if that header still contains the thread identifier that

the thread originally wrote there. Otherwise, it communicates that information to its successor thread if the thread identifier is different and thereby indicates that at least one successor is contending for access to the linked list.

77 Claims, 35 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 28

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

□ 15. Document ID: US 6170015 B1

L6: Entry 15 of 15 File: USPT Jan 2, 2001

US-PAT-NO: 6170015
DOCUMENT-IDENTIFIER: US 6170015 B1

TITLE: Network apparatus with Java co-processor

DATE-ISSUED: January 2, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Lavian; Tal I.	Sunnyvale	CA		

US-CL-CURRENT: 709/232; 709/223, 709/224, 709/235, 709/238

ABSTRACT:

A method and apparatus for automatically configuring a network switch having external network data ports, a processor, memory, data bus, and coprocessor. Network data is monitored on the external network data port. Information about the network data traffic is compared to one or more threshold conditions. The network switch is automatically configured by the coprocessor if the network data meets one of the threshold conditions. The monitor and configuration functions can be performed by software running on the coprocessor which has been downloaded from an external network maintenance station through a maintenance data port. Information about the network data traffic can be uploaded to the external network maintenance station through a maintenance data port.

16 Claims, 15 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 9

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Terms	Documents

L4 AND monitor.ab.	15

Display Format: REV Change Format

Previous Page Next Page Go to Doc#

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs

Generate OACS

Search Results - Record(s) 1 through 15 of 15 returned.

□ 1. Document ID: US 6865735 B1

L5: Entry 1 of 15 File: USPT Mar 8, 2005

US-PAT-NO: 6865735
DOCUMENT-IDENTIFIER: US 6865735 B1

TITLE: Process for rewriting executable content on a network server or desktop machine in order to enforce site specific properties

DATE-ISSUED: March 8, 2005

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Sirer; Emin Gun	Seattle	WA		
Bershad; Brian N.	Seattle	WA		

US-CL-CURRENT: 717/158; 714/38, 717/128, 718/108

ABSTRACT:

A program or program snippet is rewritten to conform to site-specific properties prior to being executed by a target host. The program or program snippet directed to a target host from a known or unknown source is either intercepted by a server before reaching the target host or can be redirected from the target host to the server to effect its rewriting. The program is parsed in its external representation, converting it to an internal representation that is inspected and analyzed with reference to a site-specific properties database. A summary of the program's properties is then compared to the site-specific properties database by a binary rewriting engine, which produces a rewritten program in an internal representation. If appropriate, the program or program snippet is rewritten to convert it to a format suitable for execution on the target host. Furthermore, certifications may be added to the rewritten program to mark that the rewritten program obeys site-specific constraints. The rewriting service thus produces a program in an appropriate target representation that conforms to site-specific properties. These properties may relate to security, auditing, optimization, monitoring, threading, and/or management of the rewritten program.

49 Claims, 6 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

□ 2. Document ID: US 6850979 B1

L5: Entry 2 of 15 File: USPT Feb 1, 2005

US-PAT-NO: 6850979
DOCUMENT-IDENTIFIER: US 6850979 B1

TITLE: Message gates in a distributed computing environment

DATE-ISSUED: February 1, 2005

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Saulpaugh; Thomas E.	San Jose	CA		
Slaughter; Gregory L.	Palo Alto	CA		
Traversat; Bernard A.	San Francisco	CA		
Abdelaziz; Mohamed M.	Santa Clara	CA		

US-CL-CURRENT: 709/225; 710/240, 713/201, 719/315

ABSTRACT:

Embodiments of message gates are described. A message gate is the message endpoint for a client or service in a distributed computing environment. A message gate may provide a secure endpoint that sends and receives type-safe messages. Gates may perform the sending and receiving of messages between clients and services using a protocol specified in a service advertisement. In one embodiment, the messages are eXtensible Markup Language (XML) messages. For a client, a message gate represents the authority to use some or all of a service's capabilities. Each capability may be expressed in terms of a message that may be sent to the service. Creation of a message gate may involve an authentication service that generates an authentication credential, and that may negotiate the desired level of security and the set of messages that may be passed between client and service. A message gate may perform verification of messages against a message schema to ensure that the messages are allowed. Message gates may embed the authentication credential in outgoing messages so that the receiving message gate may authenticate the message. Messages may also include information to allow the receiving gate to verify that the message has not been compromised prior to receipt.

40 Claims, 54 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 34

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

□ 3. Document ID: US 6792466 B1

L5: Entry 3 of 15 File: USPT Sep 14, 2004

US-PAT-NO: 6792466
DOCUMENT-IDENTIFIER: US 6792466 B1

TITLE: Trusted construction of message endpoints in a distributed computing environment

DATE-ISSUED: September 14, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Saulpaugh; Thomas E.	San Jose	CA		
Slaughter; Gregory L.	Palo Alto	CA		
Traversat; Bernard A.	San Francisco	CA		
Pouyoul; Eric	San Francisco	CA		

US-CL-CURRENT: 709/229; 707/10, 709/201, 709/203, 709/217, 709/218, 709/226, 709/227, 709/228

ABSTRACT:

In a distributed computing environment, a message gate may be the message endpoint for a client or service to communicate with another client or service. Devices may have a gate factory (e.g. message endpoint constructor) that is trusted code on the device for generating gates based on XML message descriptions. The use of the gate factory may ensure that the gate it generates is also trusted code, and that the code is correct with respect to a service advertisement. A service advertisement may indicate, for a particular service, a message schema, service URI and authentication service URI. In one embodiment, the pieces the gate factory needs to construct a gate are the XML schema of the service and the URI of the service. In another embodiment, an authentication credential may also be obtained and used in gate construction by running an authentication service specified in the service advertisement. A gate factory for a device may generate gate code that may incorporate the language, security, type safety, and/or execution environment characteristics of the local device platform. By constructing gates itself, a device has the ability to ensure that the generated gate code is relatively bug-free, produces only valid data, and provides type-safety.

43 Claims, 53 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 34

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

□ 4. Document ID: US 6789204 B2

L5: Entry 4 of 15 File: USPT Sep 7, 2004

US-PAT-NO: 6789204
DOCUMENT-IDENTIFIER: US 6789204 B2

TITLE: Resource sharing on the internet via the HTTP

DATE-ISSUED: September 7, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY

Abdelnur; Alejandro	Sunnyvale	CA
Gupta; Abhay	Milpitas	CA
Callaghan; Brent	Mountain View	CA

US-CL-CURRENT: 713/201; 709/229

ABSTRACT:

A method and apparatus for sharing resources in a network environment. An application running on a client can access a resource on a remote computer by submitting a request via an Internet browser. The request is analyzed, converted to proper format and is transferred over the network lines to a server that can satisfy the request. For security reasons, an application may not be authorized to submit a request directly to a server on the Internet. If a requesting application has a trusted status, then its request for connecting to the server is granted. If a request submitted by an application to a server is denied, then a server that entrusts the application is identified, and the request is submitted to that server. A program code called a "servlet" is implemented on that server to accept the requests submitted by a trusted application. The submitted requests are analyzed by the servlet and are forwarded to a resource server that can satisfy the requests. A response from the resource server is routed through the servlet back to the requesting application.

39 Claims, 7 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 7

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

□ 5. Document ID: US 6675382 B1

L5: Entry 5 of 15 File: USPT Jan 6, 2004

US-PAT-NO: 6675382
DOCUMENT-IDENTIFIER: US 6675382 B1

TITLE: Software packaging and distribution system

DATE-ISSUED: January 6, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Foster; Gary D.	Santa Clara	CA		

US-CL-CURRENT: 717/177; 707/1, 707/10, 717/169, 717/170, 717/172, 717/175

ABSTRACT:

A method and apparatus for packaging and distributing software. Embodiments of the invention comprise a software packaging system that is portable across many platforms. Each package is self-contained in form of a single-file entity that comprises a payload file and a control file. The payload file is an archive file that contains a compressed collection of all the software files that are required for installation of the software package. The control file includes the necessary information for installation of the files contained in the payload file, in addition to other descriptive information used to determine the size, type, location of storage, and other useful attributes of a software package, even before it is installed on a system. Security measures have been implemented in the system to detect a package the contents of which have been tampered with. Embodiments of the invention can be utilized to install packaged software that is accessible via the Internet. A package on a remote source can be accessed and installed using a Uniform Resource Locator (URL) that indicates the package's specific address on the remote source. Embodiments of the invention are designed such that the entire system is small in size so that the storage space and the transmission bandwidth required for their storage or transportation are minimized. Embodiments of the invention may be used to install, remove or update a software package.

5 Claims, 6 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 5

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

□ 6. Document ID: US 6668325 B1

L5: Entry 6 of 15 File: USPT Dec 23, 2003

US-PAT-NO: 6668325
DOCUMENT-IDENTIFIER: US 6668325 B1
**** See image for Certificate of Correction ****

TITLE: Obfuscation techniques for enhancing software security

DATE-ISSUED: December 23, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Collberg; Christian Sven	Auckland			NZ
Thomborson; Clark David	Auckland			NZ
Low; Douglas Wai Kok	Auckland			NZ

US-CL-CURRENT: 713/194; 713/200

ABSTRACT:

The present invention provides obfuscation techniques for enhancing software security. In one embodiment, a method for obfuscation techniques for enhancing software security includes selecting a subset of code (e.g., compiled source code of an application) to obfuscate, and obfuscating the selected subset of the code. The obfuscating includes applying an obfuscating transformation to the selected subset of the code. The transformed code can be weakly equivalent to the untransformed code. The applied transformation can be selected based on a desired level of security (e.g:, resistance to reverse engineering). The applied transformation can include a control transformation that can be creating using opaque constructs, which can be constructed using aliasing and concurrency techniques. Accordingly, the code can be obfuscated for enhanced software security based on a desired level of obfuscation (e.g., based on a desired potency,

resilience, and cost).

171 Claims, 55 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 27

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

□ 7. Document ID: US 6546546 B1

L5: Entry 7 of 15 File: USPT Apr 8, 2003

US-PAT-NO: 6546546
DOCUMENT-IDENTIFIER: US 6546546 B1

TITLE: Integrating operating systems and run-time systems

DATE-ISSUED: April 8, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Van Doorn; Leendert Peter	Valhalla	NY		

US-CL-CURRENT: 717/114

ABSTRACT:

The Virtual Machine is viewed by many as inherently insecure despite all the efforts to improve its security. This invention provides methods, apparatus, and computer products to implement a system that provides operating system style protection for code. Although applicable to many language systems, the invention is described for a system employing the Java language. Hardware protection domains are used to separate Java classes, provide access control on cross domain method invocations, efficient data sharing between protection domains, and memory and CPU resource control. Apart from the performance impact, these security measures are all transparent to the Java programs, even when a subclass is in one domain and its superclass is in another, when they do not violate the policy. To reduce the performance impact, classes are grouped and shared between protection domains and map data lazily as it is being shared. The system has been implemented on top of the Paramecium operating system used as an example of an extensible operating system application.

59 Claims, 7 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 7

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

□ 8. Document ID: US 6393569 B1

L5: Entry 8 of 15 File: USPT May 21, 2002

US-PAT-NO: 6393569
DOCUMENT-IDENTIFIER: US 6393569 B1

TITLE: Secured system for accessing application services from a remote station

DATE-ISSUED: May 21, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Orenshteyn; Alexander S.	Reno	NV	89509	

US-CL-CURRENT: 713/201; 709/203

ABSTRACT:

A secured system for accessing application services from at least one application program where at least one client station having low-level application independent logics stored therein and at least one controller for controlling the low-level application independent logics, the low-level application logics including a user interface logic, a device control logic for controlling devices, a file system logic, and a communication interface logic, and wherein at least one client station has means to restrict access to said application independent logics, at least one application server having high-level application logic stored in a server device for running at least one application program, the server device being coupled to said at least one application server and low-level interface between said at least one client station and said at least one server for connecting said at least one client station to said at least one application server, wherein upon accessing by said at least one client station, said at least one application server runs at least one application program which selectively controls said low-level application independent logics for controlling devices of said at least one client station and accessing data of said at least one client station without permanently storing said at least one client station data in said at least one server. There is also a description of a secure operating system and method and a secured system and method of construction of a computer system as well as description of system and method of how to preserve a running current state of an application program for security and relocation purpose.

26 Claims, 9 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 8

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw, De

☐ 9. Document ID: US 6370573 B1

L5: Entry 9 of 15 File: USPT Apr 9, 2002

US-PAT-NO: 6370573
DOCUMENT-IDENTIFIER: US 6370573 B1

TITLE: System, method and article of manufacture for managing an environment of a development architecture framework

DATE-ISSUED: April 9, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bowman-Amuah; Michel K.	Colorado Springs	CO		

US-CL-CURRENT: 709/223

ABSTRACT:

A system, method and article of manufacture are provided for managing an environment in a development architecture framework. Service of a system is managed based on service level agreements and/or operations level agreements. A plurality of system management operations are performed. The system management operations include start-up and shut-down operations, back-up and restore operations, archiving operations, security operations, and performance monitoring operations. Service is planned in order to anticipate and implement changes in the system.

15 Claims, 14 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 14

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

□ 10. Document ID: US 6324647 B1

L5: Entry 10 of 15 File: USPT Nov 27, 2001

US-PAT-NO: 6324647
DOCUMENT-IDENTIFIER: US 6324647 B1
**** See image for Certificate of Correction ****

TITLE: System, method and article of manufacture for security management in a development architecture framework

DATE-ISSUED: November 27, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bowman-Amuah; Michel K.	Colorado Springs	CO	80918	

US-CL-CURRENT: 713/201; 709/223, 713/153

ABSTRACT:

A system, method, and article of manufacture are provided for providing security management in a development architecture framework. Unauthorized attempts to access a network are detected and when an unauthorized attempt to access the network is detected, a user is notified. Access from the network is restricted to a separate wide area network. The identities of users of credit cards are verified during transactions carried out over the network. The content of electronic mail communicated over the network is also monitored so that the communication of the electronic mail over the network is prevent when the content thereof being deemed inappropriate. The electronic mail is also encrypted during the communication thereof over the network.

18 Claims, 14 Drawing figures
Exemplary Claim Number: 7
Number of Drawing Sheets: 15

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

□ 11. Document ID: US 6275938 B1

L5: Entry 11 of 15 File: USPT Aug 14, 2001

US-PAT-NO: 6275938
DOCUMENT-IDENTIFIER: US 6275938 B1

TITLE: Security enhancement for untrusted executable code

DATE-ISSUED: August 14, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bond; Barry	Renton	WA		
Bharati; Sudeep	Bellevue	WA		

US-CL-CURRENT: 713/200; 713/201, 713/202, 714/38, 714/53, 714/55

ABSTRACT:

Untrusted executable code programs (applets or controls) are written in native, directly executable code. The executable code is loaded into a pre-allocated memory range (sandbox) from which references to outside memory are severely restricted by checks (sniff code) added to the executable code. Conventional application-program interface (API) calls in the untrusted code are replaced with translation-code modules (thunks) that allow the executable code to access the host operating system, while preventing breaches of the host system's security. Static links in the code are replaced by calls to thunk modules. When an API call is made during execution, control transfers to the thunk, which determines whether the API call is one which should be allowed to execute on the operating system.

20 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

□ 12. Document ID: US 6272641 B1

L5: Entry 12 of 15 File: USPT Aug 7, 2001

US-PAT-NO: 6272641
DOCUMENT-IDENTIFIER: US 6272641 B1

TITLE: Computer network malicious code scanner method and apparatus

DATE-ISSUED: August 7, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Ji; Shuang	Santa Clara	CA		

US-CL-CURRENT: 713/201; 380/252, 713/154

ABSTRACT:

A network scanner for security checking of application programs (e.g. Java applets or Active X controls) received over the Internet or an Intranet has both static (pre-run time) and dynamic (run time) scanning. Static scanning at the HTTP proxy server identifies suspicious instructions and instruments them e.g. a pre-and-post filter instruction sequence or otherwise. The instrumented applet is then transferred to the client (web browser) together with security monitoring code. During run time at the client, the instrumented instructions are thereby monitored for security policy violations, and execution of an instruction is prevented in the event of such a violation.

27 Claims, 2 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 2

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

☐ 13. Document ID: US 6212640 B1

L5: Entry 13 of 15 File: USPT Apr 3, 2001

US-PAT-NO: 6212640
DOCUMENT-IDENTIFIER: US 6212640 B1

TITLE: Resources sharing on the internet via the HTTP

DATE-ISSUED: April 3, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Abdelnur; Alejandro	Sunnyvale	CA		
Gupta; Abhay	Milpitas	CA		
Callaghan; Brent	Mountain View	CA		

US-CL-CURRENT: 713/201

ABSTRACT:

A method and apparatus for sharing resources in a network environment. Typically, a computer linked to the Internet may have resources or may provide services, that are usable by other computers. A user, using one or more embodiments of the invention, can access those resources or services as if they were locally situated. An application running on a client can access a resource on a remote computer by

submitting a request via an Internet browser. The request is analyzed, converted to proper format and is transferred over the network lines to a server that can satisfy the request. For security reasons, an application may not be authorized to submit a request directly to a server on the Internet. For example, limitations have been implemented that prohibit a requesting application from obtaining access to resources of a server computer unless that application is a trusted application. If a requesting application has a trusted status, then its request for connecting to the server is granted. If a request submitted by an application to a server is denied, then a server that entrusts the application is identified, and the request is submitted to that server. A program code called a "servlet" is implemented on that server to accept the requests submitted by a trusted application. The submitted requests are analyzed by the servlet and are forwarded to a resource server that can satisfy the requests. A response from the resource server is routed through the servlet back to the requesting application.

51 Claims, 7 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 7

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

□ 14. Document ID: US 6092120 A

L5: Entry 14 of 15 File: USPT Jul 18, 2000

US-PAT-NO: 6092120
DOCUMENT-IDENTIFIER: US 6092120 A

TITLE: Method and apparatus for timely delivery of a byte code and serialized objects stream

DATE-ISSUED: July 18, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Swaminathan; Viswanathan	Union City	CA		
Fernando; Gerard	Mountain View	CA		
Speer; Michael	Mtn View	CA		

US-CL-CURRENT: 709/247; 709/231, 709/236

ABSTRACT:

A method and apparatus for timely delivery of classes and objects is provided. A header comprising timing information is attached to said classes and/or objects. A "start loading" time and a "load by" time are specified in the header. Other classes and/or objects to be loaded are also specified in the header. Optional compression, security, and/or error resilience schemes are also specified in the header. A process for creating the header and attaching it to a class or object is provided. A process for receiving and processing a class or object with an attached header is provided. Embodiments of the invention allow timely delivery of classes and/or objects over a wide variety of transport mechanisms, including unreliable transport mechanisms and those lacking any guarantees of timely delivery.

23 Claims, 9 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 9

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KMC | Draw. De

---

□ 15. Document ID: US 5983348 A

L5: Entry 15 of 15 File: USPT Nov 9, 1999

US-PAT-NO: 5983348
DOCUMENT-IDENTIFIER: US 5983348 A

TITLE: Computer network malicious code scanner

DATE-ISSUED: November 9, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Ji; Shuang	Santa Clara	CA		

US-CL-CURRENT: 713/200; 714/38

ABSTRACT:

A network scanner for security checking of application programs (e.g. Java applets or Active X controls) received over the Internet or an Intranet has both static (pre-run time) and dynamic (run time) scanning. Static scanning at the HTTP proxy server identifies suspicious instructions and instruments them e.g. a pre-and-post filter instruction sequence or otherwise. The instrumented applet is then transferred to the client (web browser) together with security monitoring code. During run time at the client, the instrumented instructions are thereby monitored for security policy violations, and execution of an instruction is prevented in the event of such a violation.

34 Claims, 2 Drawing figures
Exemplary Claim Number: 34
Number of Drawing Sheets: 2

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KMC | Draw. De

---

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Terms	Documents
L4 AND security.ab.	15

Display Format: REV Change Format

Previous Page Next Page Go to Doc#

Refine Search

Search Results -

Terms	Documents
L4 AND javacard	5

Database:

US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search:

```
L14
```

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Monday, March 14, 2005 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=NO; OP=OR		
L14	l4 AND javacard	5	L14
L13	L12 and (JUMP or JMP)	25	L13
L12	L11 and L10	204	L12
L11	L9 AND L3	204	L11
L10	L9 AND L4	269	L10
L9	717/$$$.cclsOR 713/$$$.ccls.	13568	L9
L8	L7 AND L6	0	L8
L7	L5 AND monitor.ab.	15	L7
L6	L5 AND security.ab.	15	L6
L5	L4 and ((byte ADJ code) OR (bytecode))	525	L5
L4	java and monitor	4280	L4
L3	java and security and monitor	1831	L3
L2	java near security near monitor	0	L2

L1	6092120.pn. or 5974549.pn. or 6023764.pn. or 6199181.pn. or 6275938.pn. or 6802054.pn. or 6557168.pn. or 6557168.pn. or 6510352.pn. or 6327700.pn. or 6668325.pn. or 6546546.pn. or 5983348.pn.	12	L1

END OF SEARCH HISTORY

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 12 of 12 returned.

☑ 1. Document ID: US 6802054 B2

L1: Entry 1 of 12 File: USPT Oct 5, 2004

US-PAT-NO: 6802054
DOCUMENT-IDENTIFIER: US 6802054 B2

TITLE: Generation of runtime execution traces of applications and associated problem determination

DATE-ISSUED: October 5, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Faraj; Mazen	North York			CA

US-CL-CURRENT: 717/128; 714/45

ABSTRACT:

A computer system for generating and analyzing application trace data includes a monitor for launching Java language virtual machines using the Java Platform Debug Architecture to enable the virtual machines to generate event data on the occurrence of specified events during application execution on the virtual machines. The event data is placed on an event queue and the monitor removes the event data from the event queue for forwarding to a logging service. The logging service records the event data in a trace file. A set of problem determination tools use defined product description, and the trace file data to provide an analysis to a user based on a defined level of analysis selected by the user from product, component, code and logical levels of analysis.

25 Claims, 1 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 1

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☑ 2. Document ID: US 6668325 B1

L1: Entry 2 of 12 File: USPT Dec 23, 2003

US-PAT-NO: 6668325
DOCUMENT-IDENTIFIER: US 6668325 B1
**** See image for Certificate of Correction ****

TITLE: Obfuscation techniques for enhancing software security

DATE-ISSUED: December 23, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Collberg; Christian Sven	Auckland			NZ
Thomborson; Clark David	Auckland			NZ
Low; Douglas Wai Kok	Auckland			NZ

US-CL-CURRENT: 713/194; 713/200

ABSTRACT:

The present invention provides obfuscation techniques for enhancing software security. In one embodiment, a method for obfuscation techniques for enhancing software security includes selecting a subset of code (e.g., compiled source code of an application) to obfuscate, and obfuscating the selected subset of the code. The obfuscating includes applying an obfuscating transformation to the selected subset of the code. The transformed code can be weakly equivalent to the untransformed code. The applied transformation can be selected based on a desired level of security (e.g., resistance to reverse engineering). The applied transformation can include a control transformation that can be creating using opaque constructs, which can be constructed using aliasing and concurrency techniques. Accordingly, the code can be obfuscated for enhanced software security based on a desired level of obfuscation (e.g., based on a desired potency, resilience, and cost).

171 Claims, 55 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 27

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

---

☑ 3. Document ID: US 6557168 B1

L1: Entry 3 of 12 File: USPT Apr 29, 2003

US-PAT-NO: 6557168
DOCUMENT-IDENTIFIER: US 6557168 B1

TITLE: System and method for minimizing inter-application interference among static synchronized methods

DATE-ISSUED: April 29, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Czajkowski; Grzegorz J.	Mountain View	CA		

US-CL-CURRENT: 717/151; 713/375, 717/127

ABSTRACT:

A system and method for isolating the execution of a plurality of applications. A plurality of monitors are provided for a plurality of applications to access a static synchronized method. The applications are enabled to call the static synchronized method concurrently by accessing the static synchronized method through the plurality of monitors. A plurality of threads within one of the applications are excluded from calling the static synchronized method concurrently. The source code or bytecode for the synchronized method may be transformed by removing a method-level monitor and adding the plurality of monitors inside the method. In one embodiment, each static synchronized method is replaced with a corresponding static non-synchronized method. The applications may be further isolated by placing the static fields of shared classes into a static field class, which has one instance per utilizing application. The static non-synchronized method includes the body of the corresponding static synchronized method, wherein the body is synchronized on the instance of the static field class that corresponds to the utilizing application.

30 Claims, 11 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 11

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

---

☑ 4. Document ID: US 6546546 B1

L1: Entry 4 of 12 File: USPT Apr 8, 2003

US-PAT-NO: 6546546
DOCUMENT-IDENTIFIER: US 6546546 B1

TITLE: Integrating operating systems and run-time systems

DATE-ISSUED: April 8, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Van Doorn; Leendert Peter	Valhalla	NY		

US-CL-CURRENT: 717/114

ABSTRACT:

The Virtual Machine is viewed by many as inherently insecure despite all the efforts to improve its security. This invention provides methods, apparatus, and computer products to implement a system that provides operating system style protection for code. Although applicable to many language systems, the invention is described for a system employing the Java language. Hardware protection domains are used to separate Java classes, provide access control on cross domain method invocations, efficient data sharing between protection domains, and memory and CPU resource control. Apart from the performance impact, these security measures are all transparent to the Java programs, even when a subclass is in one domain and its superclass is in another, when they do not violate the policy. To reduce the performance impact, classes are grouped and shared between protection domains and map data lazily as it is being shared. The system has been implemented on top of the Paramecium operating system used as an example of an extensible operating system application.

59 Claims, 7 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 7

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

☑ 5. Document ID: US 6510352 B1

L1: Entry 5 of 12 File: USPT Jan 21, 2003

US-PAT-NO: 6510352
DOCUMENT-IDENTIFIER: US 6510352 B1

TITLE: Methods and apparatus for object-based process control

DATE-ISSUED: January 21, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Badavas; Paul C.	Southboro	MA		
Hansen; Peter D.	Wellesley	MA		

US-CL-CURRENT: 700/19; 700/18, 700/20, 700/48, 700/49, 700/50, 700/52, 718/100

ABSTRACT:

The provides improved control devices, systems and methods for operation thereof. These rely on control devices that provide virtual machine environments in which Java objects, or other such software constructs, are executed to implement control (e.g., to monitor and/or control a device, process or system). These objects define blocks which are the basic functional unit of the control. They also define the input, output and body parts from which blocks are formed, and the signals that are communicated between blocks. The objects also define nested and composite groupings of blocks used to control loops and higher-level control functions.

72 Claims, 18 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 18

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

☑ 6. Document ID: US 6327700 B1

L1: Entry 6 of 12 File: USPT Dec 4, 2001

US-PAT-NO: 6327700
DOCUMENT-IDENTIFIER: US 6327700 B1

TITLE: Method and system for identifying instrumentation targets in computer programs related to logical transactions

DATE-ISSUED: December 4, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Chen; J. Bradley	Seattle	WA		
Bershad; Brian N.	Seattle	WA		

US-CL-CURRENT: 717/127; 717/130

ABSTRACT:

A method and system for identifying sets of instructions within a computer program, execution of which serve as an indicator for processing of a transaction by the computer program and that together comprise a witness set. The witness set may be employed to monitor execution of the computer program and detect processing of the transaction. Witness sets are constructed by iteratively filtering an initial set of instructions based on profile data collected during execution of the computer program.

28 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. De

☑ 7. Document ID: US 6275938 B1

L1: Entry 7 of 12 File: USPT Aug 14, 2001

US-PAT-NO: 6275938
DOCUMENT-IDENTIFIER: US 6275938 B1

TITLE: Security enhancement for untrusted executable code

DATE-ISSUED: August 14, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Bond; Barry	Renton	WA		
Bharati; Sudeep	Bellevue	WA		

US-CL-CURRENT: 713/200; 713/201, 713/202, 714/38, 714/53, 714/55

ABSTRACT:

Untrusted executable code programs (applets or controls) are written in native, directly executable code. The executable code is loaded into a pre-allocated memory range (sandbox) from which references to outside memory are severely restricted by checks (sniff code) added to the executable code. Conventional application-program

interface (API) calls in the untrusted code are replaced with translation-code modules (thunks) that allow the executable code to access the host operating system, while preventing breaches of the host system's security. Static links in the code are replaced by calls to thunk modules. When an API call is made during execution, control transfers to the thunk, which determines whether the API call is one which should be allowed to execute on the operating system.

20 Claims, 4 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

☑ 8. Document ID: US 6199181 B1

L1: Entry 8 of 12 File: USPT Mar 6, 2001

US-PAT-NO: 6199181
DOCUMENT-IDENTIFIER: US 6199181 B1
**** See image for Certificate of Correction ****

TITLE: Method and system for maintaining restricted operating environments for application programs or operating systems

DATE-ISSUED: March 6, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Rechef; Eran	Lehavim			IL
Raanan; Gil	Zoran			IL
Solan; Eilon	Herzlia			IL

US-CL-CURRENT: 714/38; 713/201

ABSTRACT:

A method for protecting an operating environment on a processor from a rogue program operating on the processor comprising isolating simultaneously executing programs or operating systems is disclosed. Memory space for use only by the first program while the first program is executing is allocated. Communication between the first program and the computer's operating environment is accomplished through a single link employing one of several methods including using shared memory space, a dedicated interrupt or a dedicated I/O port. The monitor manages a restricted operating environment for the first program on the processor, the restricted operating environment preventing the first program from accessing resources on the processor except for the allocated memory space the single communication link.

21 Claims, 10 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 9

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

☑ 9. Document ID: US 6092120 A

L1: Entry 9 of 12 File: USPT Jul 18, 2000

US-PAT-NO: 6092120
DOCUMENT-IDENTIFIER: US 6092120 A

TITLE: Method and apparatus for timely delivery of a byte code and serialized objects stream

DATE-ISSUED: July 18, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Swaminathan; Viswanathan	Union City	CA		
Fernando; Gerard	Mountain View	CA		
Speer; Michael	Mtn View	CA		

US-CL-CURRENT: 709/247; 709/231, 709/236

ABSTRACT:

A method and apparatus for timely delivery of classes and objects is provided. A header comprising timing information is attached to said classes and/or objects. A "start loading" time and a "load by" time are specified in the header. Other classes and/or objects to be loaded are also specified in the header. Optional compression, security, and/or error resilience schemes are also specified in the header. A process for creating the header and attaching it to a class or object is provided. A process for receiving and processing a class or object with an attached header is provided. Embodiments of the invention allow timely delivery of classes and/or objects over a wide variety of transport mechanisms, including unreliable transport mechanisms and those lacking any guarantees of timely delivery.

23 Claims, 9 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 9

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

☑ 10. Document ID: US 6023764 A

L1: Entry 10 of 12 File: USPT Feb 8, 2000

US-PAT-NO: 6023764
DOCUMENT-IDENTIFIER: US 6023764 A

TITLE: Method and apparatus for providing security certificate management for Java Applets

DATE-ISSUED: February 8, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Curtis; Bryce Allen	Round Rock	TX		

US-CL-CURRENT: 713/200

ABSTRACT:

The present invention defines a means for establishing a secure connection between a Java Applet and a secure web server for protocols other than Https via the use of a Java Security Service. More specifically, the present invention uses the web browser's installed certificates to setup and establish an encrypted session between the Java Applet and the secure web server. The secure connection is then used to retrieve the certificates required by the Java security service.

15 Claims, 5 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 4

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☑ 11. Document ID: US 5983348 A

L1: Entry 11 of 12 File: USPT Nov 9, 1999

US-PAT-NO: 5983348
DOCUMENT-IDENTIFIER: US 5983348 A

TITLE: Computer network malicious code scanner

DATE-ISSUED: November 9, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Ji; Shuang	Santa Clara	CA		

US-CL-CURRENT: 713/200; 714/38

ABSTRACT:

A network scanner for security checking of application programs (e.g. Java applets or Active X controls) received over the Internet or an Intranet has both static (pre-run time) and dynamic (run time) scanning. Static scanning at the HTTP proxy server identifies suspicious instructions and instruments them e.g. a pre-and-post filter instruction sequence or otherwise. The instrumented applet is then transferred to the client (web browser) together with security monitoring code. During run time at the client, the instrumented instructions are thereby monitored for security policy violations, and execution of an instruction is prevented in the event of such a violation.

34 Claims, 2 Drawing figures
Exemplary Claim Number: 34
Number of Drawing Sheets: 2

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

☑ 12. Document ID: US 5974549 A

L1: Entry 12 of 12 File: USPT Oct 26, 1999

US-PAT-NO: 5974549
DOCUMENT-IDENTIFIER: US 5974549 A

TITLE: Security monitor

DATE-ISSUED: October 26, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Golan; Gilad	Ramat Hasharon			IL

US-CL-CURRENT: 713/200; 714/47

ABSTRACT:

The present invention is a method of creating a secure sandbox within which a plurality of downloaded software components can execute in a secure manner. The software components can be of any type, e.g., Java, ActiveX, Netscape plugin, etc. The invention implements a security monitor that is injected to the address space of an arbitrary monitored application such as a Web browser, e.g., Internet Explorer, Netscape Navigator, etc. The monitored application then executes in a secure mode in which every software component downloaded executes in a secure sandbox. The security monitor detects when such a software component is downloaded and is operative to create the sandbox around it before it is permitted to execute. If the software component attempts to commit an action that breaches security, it halts the software component's execution and issues a warning to the user. The security monitor detects attempted security breaches by the software component in accordance with a user configurable security policy. Such a policy may include limiting file read/write access, access to directories, disk access, creation and the reading/writing of network connections, access to system resources and services and access to the address spaces of other processes.

19 Claims, 15 Drawing figures
Exemplary Claim Number: 12
Number of Drawing Sheets: 14

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Terms	Documents
6092120.pn. or 5974549.pn. or 6023764.pn. or 6199181.pn. or	

6275938.pn. or 6802054.pn. or 6557168.pn. or 6557168.pn. or 6510352.pn. or 6327700.pn. or 6668325.pn. or 6546546.pn. or 5983348.pn.	12

Display Format: REV Change Format

Previous Page Next Page Go to Doc#